CommWorks
a 3Com company

FAS Server

High Level and Detailed Design Specification

Wireless NextGen

PPS ID: 1186 Feature ID: FAS Server

Feature Activation System – Server

Approvers:

TITLE	NAME	SIGNATURE & DATE
Product Management	Lee Rosenbaum	
Engineering Lead	Ching Kung	
Program Manager	Carolyn Heide	
Architect	Christian Rigg	

RECORD OF REVISIONS			
ORIGINATOR:	REVISED SECTION/PARAGRAPH:	REVISION LEVEL:	UPDATED DATE:
Senthil M Sundaram	Initial Document	0.1	XXXXXXX
Ramesh Naidu	6.6 CLI Thread Component 6.7 FASFeatureFileLoaderThread Component	0.2	XXXXXXX
Senthil M Sundaram	Section 10 & 11 + Review updates	0.3	XXXXXXX
Senthil M Sundaram	Section 10 & 11 minor updates	1.0	XXXXXXX
Ramesh Naidu	Updated Appendix A, B, C	1.1	XXXXXXX
Senthil M Sundaram	Updated Design Review comments and Added Appendix F	1.1	XXXXXXX
Senthil M Sundaram	Appendix F – FASServer MIB Appendix E – FAS Protocol Changes	1.2	XXXXXXX
Senthil M Sundaram	Changed Approval List, Updated Appendix F – FASServer MIB	1.3	XXXXXXX
Senthil M Sundaram	Changed FeatureID type from TEXT to INTEGER. Changed some of the class names Section 11.2 - Changed Tag numbering.	1.4	XXXXXXX

Table of Contents

Table of Figures

1. INTRODUCTION

1.1 Scope

This document provides a detail design for Feature Activation Server, which is a component of Feature Activation System. The FAS Server component is intended to provide management and distribution of feature keys to NEs. This software is distributed along with the CEM software package as an optional component.

The notation used in the documents is self-explanatory and in some places it is specified using Unified Modeling Language (UML) deployment diagram, Class diagrams and activity diagrams.

1.2 Overview

The FAS Server is intended to facilitate ease of feature deployment. The primary method of achieving this is based on the use of Network Feature Keys. A network Feature Key exists only because the Server is capable of converting it into an Element Feature Key as required by the Feature Activation Agent running on elements to be activated.

The fundamental architectural model calls for the FAS Agent to Pull Activations from the FAS Server. The FAS Server will provide these Activations and report successes, failures and current status. These functions will be performed via existing and proprietary protocols.

There is one FAS Server per customer network management domain (FAS Domain). It can reside on the same machine as the CEM Server.

1.3 External Interfaces

The external interfaces supported by FAS Server are

- *Feature Key file loading*

  This facilitates loading of Network Feature Keys into FAS Server thro' feature key files. These files are populated into the "FAS Server directory" using FTP/TFTP

- *FAS Protocol*

  TCP/IP based FAS Protocol is used to communicate between FAS Agent and FAS Server

- *Command Line interface (CLI)*

  CLI facilitates user to execute commands and get the result in the Command Window provided by FAS Server

- *SNMP*

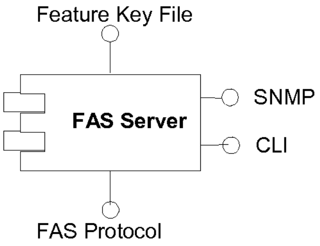


Figure 1 FAS Interface Diagram

1.4 Network Architecture

The deployment of FAS Server in the customer network looks like this.

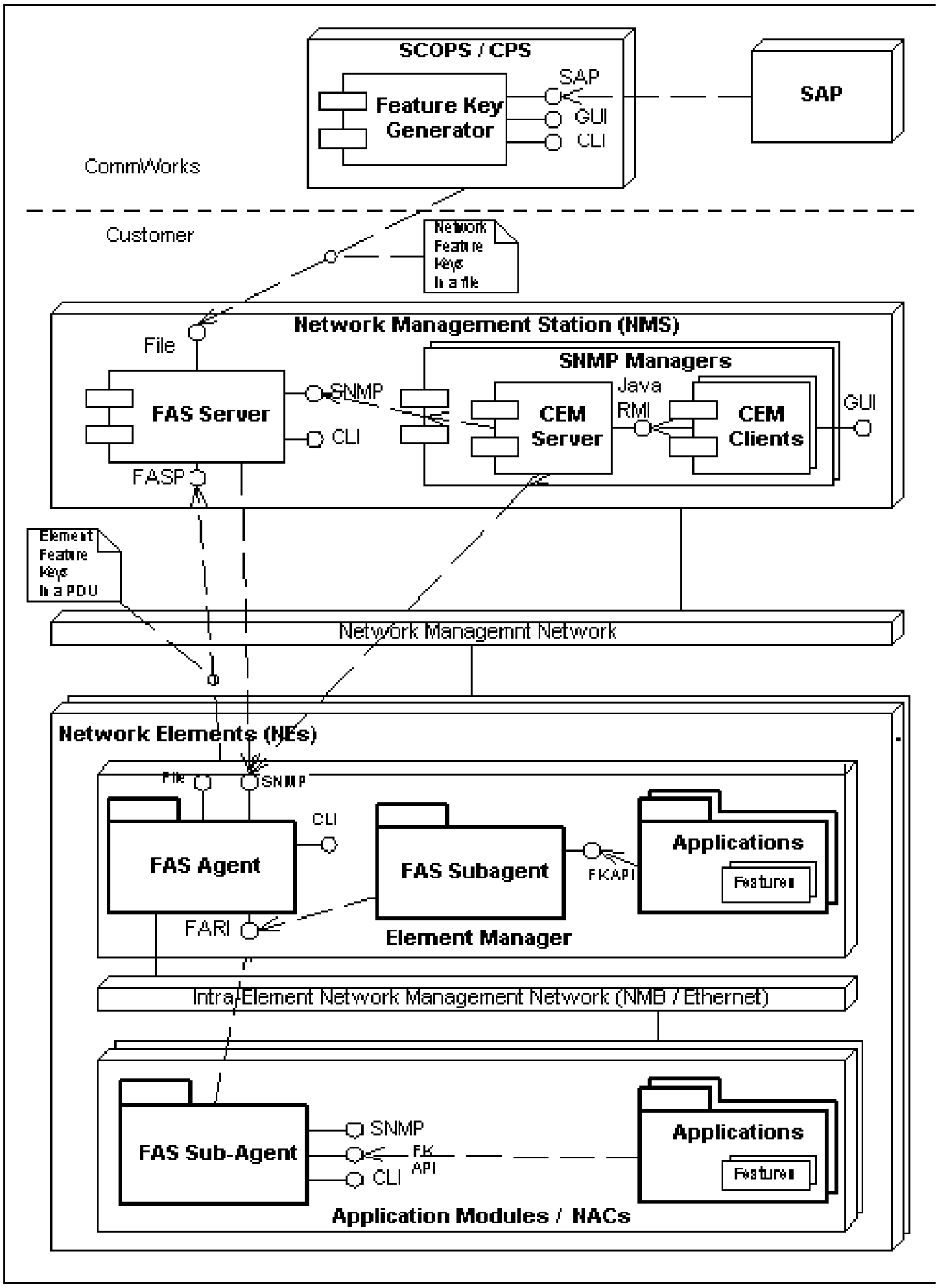


Figure 2 FAS Server network diagram

2. APPLICABLE DOCUMENTS

This section shall list the number, title, revision and date of all documents referenced in this Detailed Software Design Specification. This section should include the list below as well as any other applicable documents (i.e. international standards, federal standards, etc).

Title
FAS 30142 SysFS
FAS Agent Sw FS
FAS Sub-Agent FS
Feature ID 56 FFD
FS_FAS_FeatureKeyGenerator

3. ACRONYMS

AM	Application Module
CEM	Common Element Manager
CLI	Command Line Interface
CPS	CommWorks Professional Services
FAS	Feature Activation System
FASS	FAS Server
FASP	FAS Protocol
FFD	Feature Functional Document
GUI	Graphical User Interface
IPSec	Internet Protocol Security
MIB	Management Information Base
NE	Network Element
NAC	Network Access Card
NMC	Network Management Card
NMS	Network Management System
QOS	Quality of Service
SAP	The largest integrated enterprise software company. Also refers to the software system provided by SAP. SAP provides integrated software for accounting, finance, inventory, supply chain management, customer relations management, supplier relations management, HR management and so on. SAP software is used to manage the CommWorks supply chain.
SC	Shelf Controller
SCOPS	Supply Chain Operations
SM	System Manager
SNMP	Simple Network Management Protocol
SPI	Service Provider Interface
TC2000	Total Control 2000
TCP	Transmission Control Protocol
XML	Extensible Markup Language

4. DEFINITIONS

Feature Activation System (FAS) – Feature Activation System consists of having a feature key to communicate what features are permitted to be activated and having processing entities that act on the activation request. The feature activation process can be envisioned as being handled by a combination of processing entities each of which has certain responsibilities. A part of the Feature Activation System must be present in each Network Element in order to allow the customer to apply activation without reliance on any system external to the Network Element.

FAS Server - This is the component of the Network Management Station that distributes Feature Keys to the Network Elements. There is one FAS Server per customer network management domain (FAS Domain). It can reside on the same machine as the CEM Server. The FAS Server software is distributed together with the CEM software package.

FAS Agent – This is the subsystem in the Network Element that receives and stores Feature Keys. It divides and distributes Feature Keys to the Application Modules. It resides in the Network Element Manager (System Manager in Total Control 2000 system) part of the NE.

FAS Sub-agent – This is the subsystem in the Application Module that receives Feature Keys from the FAS Agent. In Total Control 2000 system, AM-agents shall have FAS-Subagent.

CEM - This is the SNMP manager that communicates to the SNMP Agents such as SM, FAS-Server, FAS-Agents, etc.

Feature Key Generator - This is the component that generates Feature Keys. It is operated by CommWorks. A database is needed for storing Feature Key specific information. There is no direct communications between the Feature Key Generator and the customer's network.

Feature Identifier – This is the identifier for a particular feature category, like QOS, IPSEC, etc.

Feature Key – This is a piece of data that is used by the FAS Server and the FAS Agent to activate a feature.

Feature Serial Number – This is a serial number embedded in the feature key that gets generated by the Feature Key Generator at CommWorks. The feature key serial number is maintained by CommWorks for each feature identifier issued. FAS-Server propagates the feature key serial number from Network Key to the Element Keys.

Feature Unit (FU) - One *Feature Unit* is the elementary unit (permission) of a feature. One Feature Unit is needed to activate one Feature Category on one FAS Sub-agent. The Feature Key includes a [1 to X] number of Feature Units (permissions).

Network Feature Key – Feature key(s) generated by the CommWorks Feature Key Generator in XML file format for the FAS Server. This is based on the serial number/host id of the machine where FAS Server is running. This is generated for customers.

Element Feature Key - Feature key(s) generated by the CommWorks Feature Key Generator or by FAS Server in XML file format. These are used by the FAS Agent (System Manager in Total Control 2000 system). This is based on the serial number of the control shelf in Total Control 2000 system. These keys are generated automatically by the FAS Server from the Network Feature Keys, or by CommWorks using the Feature Key Generator if the serial number of the control shelf is available.

System Manager (SM) - The System Manager provides the single point of network management for the Total Control 2000 system. It is responsible for storage and distribution of software loads and initial configuration for all the Total Control 2000 application blades. The System Manager also provides the SCB interconnects for supporting multi-shelf systems. SM has the FAS-Agent.

5. REQUIREMENTS

This section contains the requirements implemented at each version of FAS Server

5.1 FAS Server Version 1.0

These are requirements to be implemented in FAS Server Version 1.0

5.1.1 Implement FAS Protocol (TCP based communication protocol between FAS Agent and FAS Server.

5.1.2 The FAS Server incrementally learns about NEs when received Feature Key request.

5.1.3 FAS Server adds NE to its NE inventory based on NE identifier.

5.1.4 Loading Network Feature Keys into the FAS Server via Feature Key Files. All the Feature Key Files shall be loaded from a specified directory.

5.1.5 The FAS Server shall keep track in persistent storage of every Feature Key received from CommWorks that is valid for that FAS Server.

5.1.6 The FAS Server should check the Feature Keys for duplication.

5.1.7 The FAS Server should be able to re-use Feature Keys.

5.1.8 FAS Server should support unloading of Feature Key.

5.1.9 FAS Server creates an Element Feature Key based on NE identifier (Serial number) and Network Feature Key. It also remembers Element Feature Keys that it created

5.1.10 FAS server can reclaim a previously allocated key, the steps required for this operation may include

- Delete a key on the NE.

- NE Sends a Feature Request to FAS Server with lesser number of feature Unit than it has.

- FAS Server Adds the claimed key to feature pool.

5.1.11 The FAS Server should support a log.

5.1.12 The FAS Server should verify the Network Feature Keys and authenticate the Feature Key Generator as the originator.

5.1.13 FAS Server should support Command Line Interface (CLI).

5.1.14 FAS Server should support SNMP Interface.

6. DESIGN

The architecture of the system is described in terms of the components that are identified and their interactions. The FAS Server is part of the Feature Activation System that performs feature key management and distribution. The server is designed using Object Oriented technique.

6.1 Classification of components and classes

The following list of components has been identified during the design of the architecture. It is defined very briefly with just enough detail to explain how they interact in order to provide the main required functionality. FAS Server components are

- FAS Server Implementation Objects
    - FAS Server Thread
    - FAS Feature File Loader Thread
    - FAS Connect Handler Thread
    - FAS Request Handler Thread
    - FAS CLI Thread
- Platform Objects
- Communication Objects
- Authentication Objects

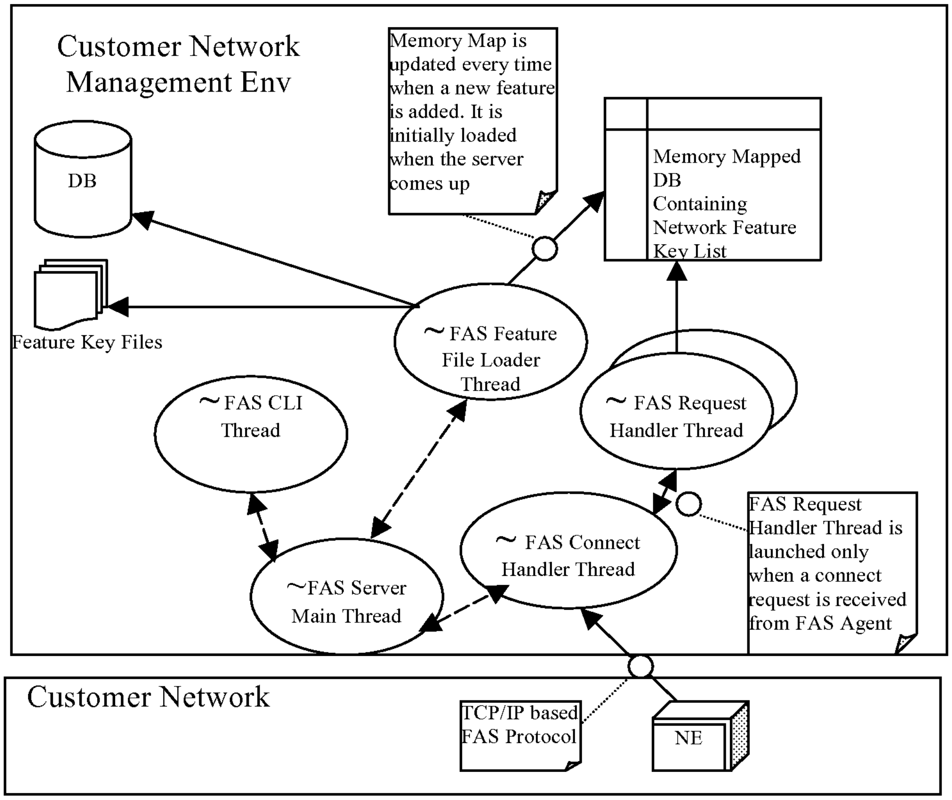

Customer Network Management Env
Memory Map is updated every time when a new feature is added. It is initially loaded when the server comes up
DB
Memory Mapped DB Containing Network Feature Key List
Feature Key Files
~ FAS Feature File Loader Thread
~ FAS CLI Thread
~ FAS Request Handler Thread
FAS Request Handler Thread is launched only when a connect request is received from FAS Agent
~ FAS Connect Handler Thread
~FAS Server Main Thread
Customer Network
TCP/IP based FAS Protocol
NE

Figure 3 FAS Server Architecture diagram

Platform object consist of classes pertaining to log implementation/maintenance, configuration, parsing the XML file and parsing request packets.

Communication object provides communication infrastructure used to interface with external systems like DB, FASAgents etc.

Authentication object contains classes that performs authentication between FAS Key Generator & FAS Server and between FAS Server & FAS Agent. FAS Key Generator to FAS Server authentication is done using the auth string present along with the Feature key. FAS Server to FAS Agent authentication is performed using the auth string embedded into the FAS Protocol packet.

FAS Server implementation Objects are the actual implementations of FAS Server that does feature key management and distribution. All runnable objects are implemented in this package. These objects uses the facilities provided by other object classifications.

6.2 Message Flow Sequence

This section describes the message flow sequences between FAS Subagent, FAS Agent & FAS Server components of FAS system.

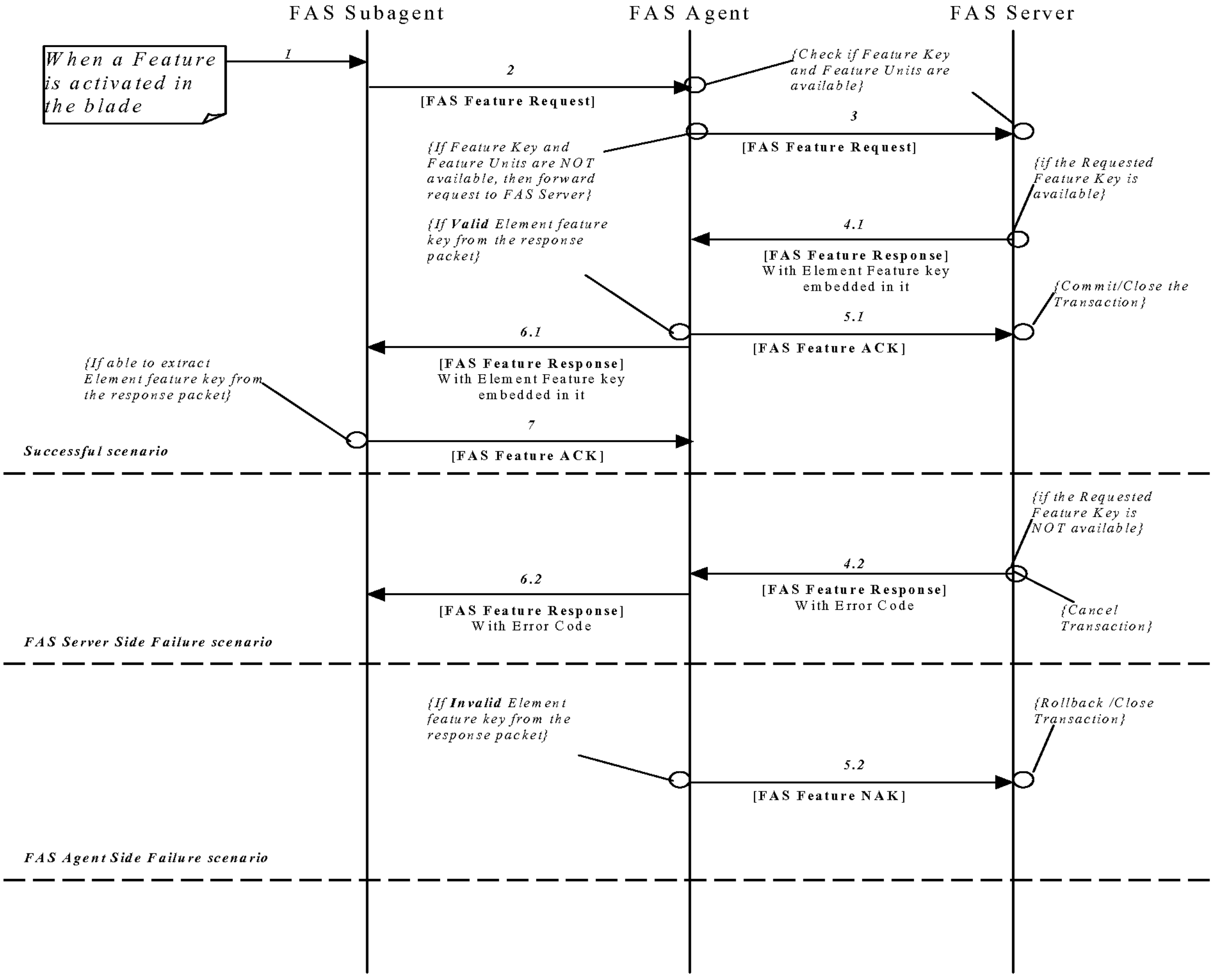

FAS Subagent
FAS Agent
FAS Server
When a Feature is activated in the blade
1
2
[FAS Feature Request]
{Check if Feature Key and Feature Units are available}
3
[FAS Feature Request]
{If Feature Key and Feature Units are NOT available, then forward request to FAS Server}
{if the Requested Feature Key is available}
{If Valid Element feature key from the response packet}
4.1
[FAS Feature Response] With Element Feature key embedded in it
{Commit/Close the Transaction}
5.1
[FAS Feature ACK]
6.1
[FAS Feature Response] With Element Feature key embedded in it
{If able to extract Element feature key from the response packet}
7
[FAS Feature ACK]
Successful scenario
{if the Requested Feature Key is NOT available}
4.2
[FAS Feature Response] With Error Code
6.2
[FAS Feature Response] With Error Code
{Cancel Transaction}
FAS Server Side Failure scenario
{If Invalid Element feature key from the response packet}
{Rollback /Close Transaction}
5.2
[FAS Feature NAK]
FAS Agent Side Failure scenario

Figure 4 Message Flow Sequence Diagram

6.3 Class Interaction

Having introduced the decomposition of the system in the previous section. This section provides the class interaction details for all the major classes identified.

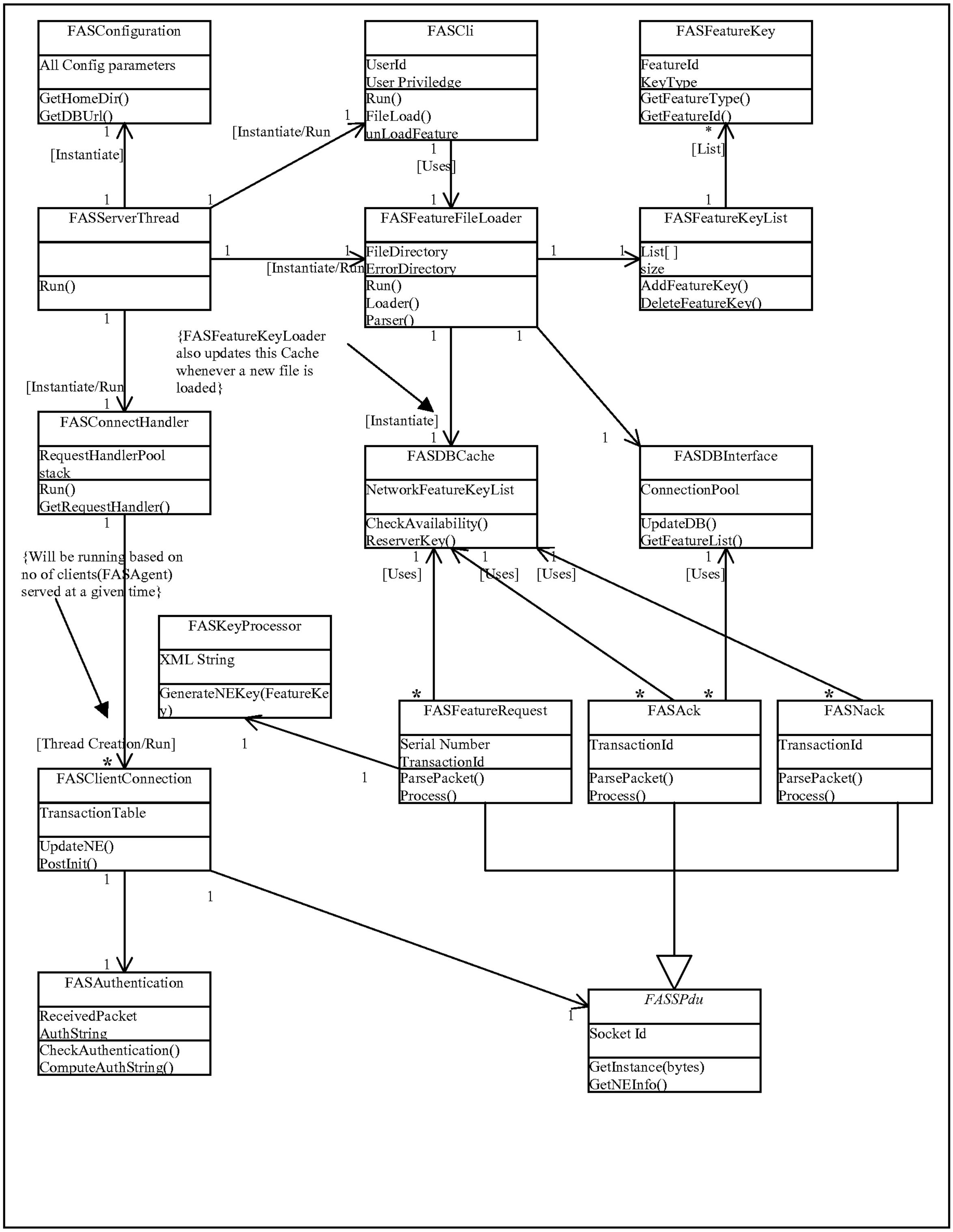


Figure 5 Class diagram

6.4 FASServerThread Component

6.4.1 Purpose

FAS Server Thread is the main component that launches and controls all other threads and objects created in the server environment.

6.4.2 Requirements/Functionality

The functionality and requirements implemented in this component are

- The FAS Server Shall support a log. (Requirement 5.1.11)
- Launching of all the runnable components
- Loading Configuration parameters
- Establishing DB Connections
- Loading of feature keys into Memory Mapped DB

6.4.3 Input/Output

Configuration File (Input)

6.4.4 Processing

Processing of FAS Server thread is shown below with an activity diagram.

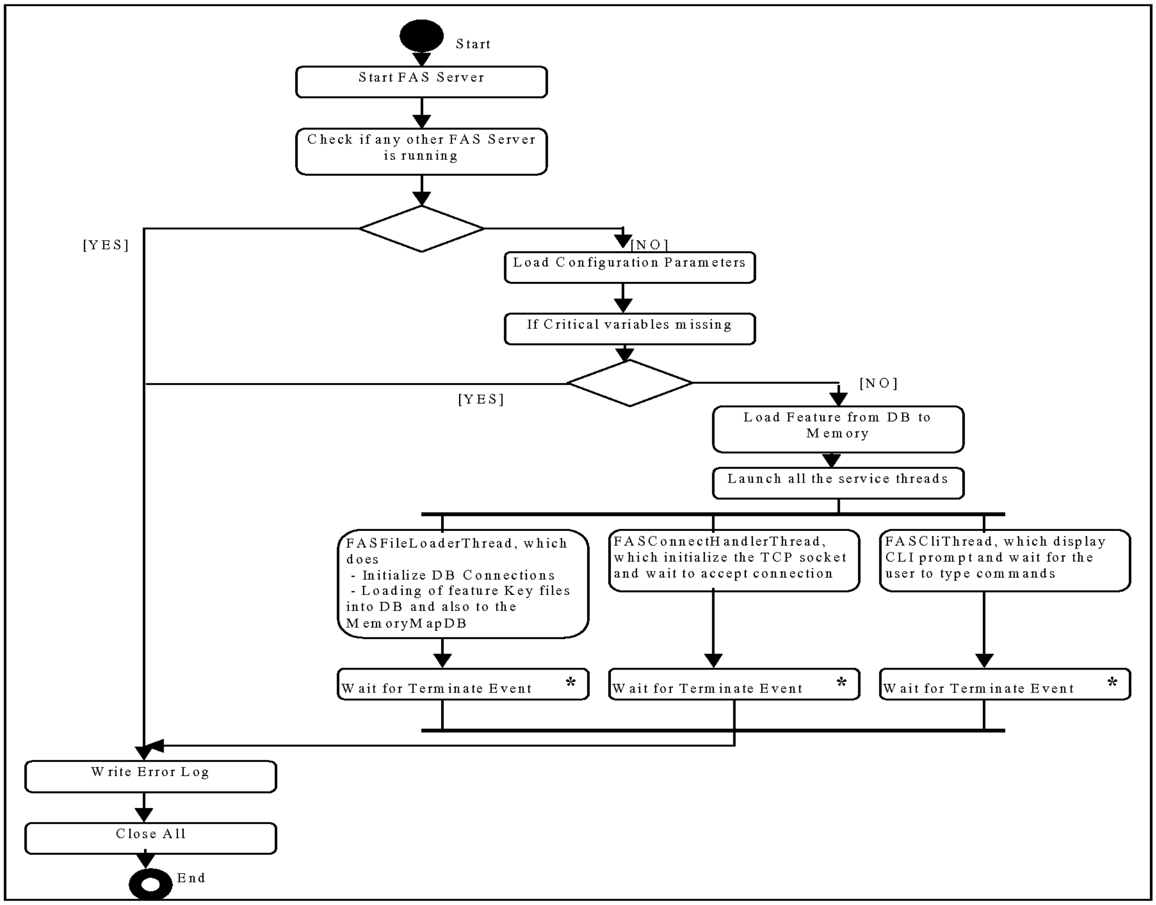

Start
Start FAS Server
Check if any other FAS Server is running
[YES]
[NO]
Load Configuration Parameters
If Critical variables missing
[YES]
[NO]
Load Feature from DB to Memory
Launch all the service threads
FASFileLoaderThread, which does
- Initialize DB Connections
- Loading of feature Key files into DB and also to the MemoryMapDB
FASConnectHandlerThread, which initialize the TCP socket and wait to accept connection
FASCliThread, which display CLI prompt and wait for the user to type commands
Wait for Terminate Event *
Wait for Terminate Event *
Wait for Terminate Event *
Write Error Log
Close All
End

Figure 6 Activity diagram for FASServer Thread

6.5 FASConnectHandlerThread Component

6.5.1 Purpose

FAS Connect Handler Thread accepts connect requests from FASAgents and spans FASClientConnectionThread from the pool of FASClientConnectionThread to serve the client request.

6.5.2 Requirements/Functionality

The functionality and requirements implemented in this component are

- FAS Server shall support a log. (Requirement 5.1.11)
- Create and maintenance of FASClientConnectionThread Pool
- Launching of FASClientConnectionThread

6.5.3 Input/Output

FASAgent Connect Request (Input)

FAS Request Handler Thread (Output)

6.5.4 Processing

FASConnectHandlerThread processing is shown in diagram below.

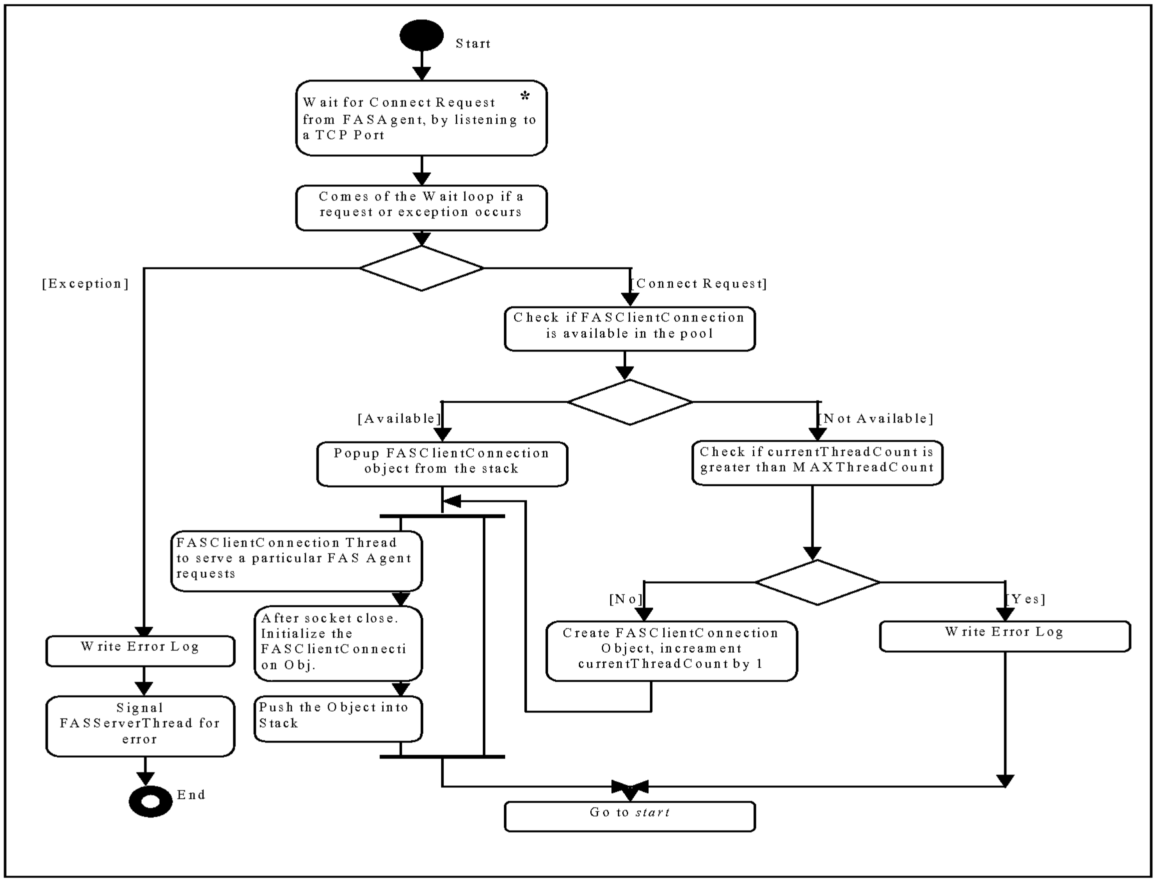

Start
Wait for Connect Request from FASAgent, by listening to a TCP Port
*
Comes of the Wait loop if a request or exception occurs
[Exception]
[Connect Request]
Check if FASClientConnection is available in the pool
[Available]
[Not Available]
Popup FASClientConnection object from the stack
Check if currentThreadCount is greater than MAXThreadCount
FASClientConnection Thread to serve a particular FAS Agent requests
After socket close. Initialize the FASClientConnection Obj.
Push the Object into Stack
[No]
[Yes]
Create FASClientConnection Object, increament currentThreadCount by 1
Write Error Log
Write Error Log
Signal FASServerThread for error
End
Go to start

Figure 7 Activity Diagram for FASConnectHandlerThread

6.6 FASClientConnectionThread Component

6.6.1 Purpose

FAS Request Handler Thread receives messages from FAS Agent, process it and sends response back to the FASAgent based on the request type received. A single thread will serve only one FAS Agent.

6.6.2 Requirements/Functionality

The functionality and requirements implemented in this component are

- Implement FAS Protocol (TCP based communication protocol between FAS Agent and FAS Server) (Requirement 5.1.1).
- FAS Server incrementally learns about NEs when told to activate features on NEs. (Requirement 5.1.2)
- FAS Server adds NE to its NE inventory based on NE identifier. (Requirement 5.1.3)
- FAS Server creates an Element Feature Key based on NE identifier and Network Feature Key. It will also remember Element Feature Keys that is created. (Requirement 5.1.5)

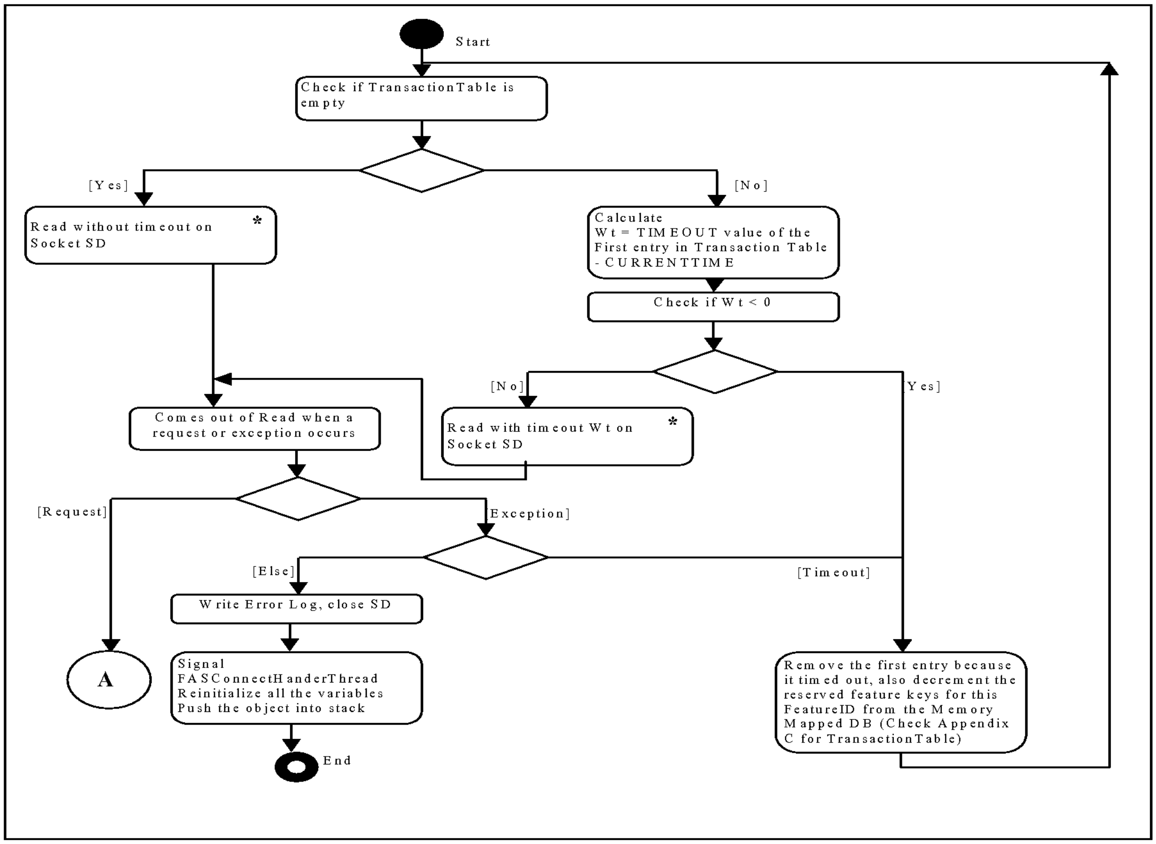
6.6.3 Input/Output

Request from FAS Agent (Input)

Response generated to FAS Agent (Output)

6.6.4 Processing

FASClientConnectionThread processing is shown in diagram below.


Start
Check if TransactionTable is empty
[Yes]
[No]
Read without timeout on Socket SD *
Calculate
Wt = TIMEOUT value of the First entry in Transaction Table - CURRENTTIME
Check if Wt < 0
[No]
Yes]
Read with timeout Wt on Socket SD *
Comes out of Read when a request or exception occurs
[Request]
Exception]
[Else]
[Timeout]
Write Error Log, close SD
A
Signal FASConnectHanderThread
Reinitialize all the variables
Push the object into stack
End
Remove the first entry because it timed out, also decrement the reserved feature keys for this FeatureID from the Memory Mapped DB (Check Appendix C for TransactionTable)

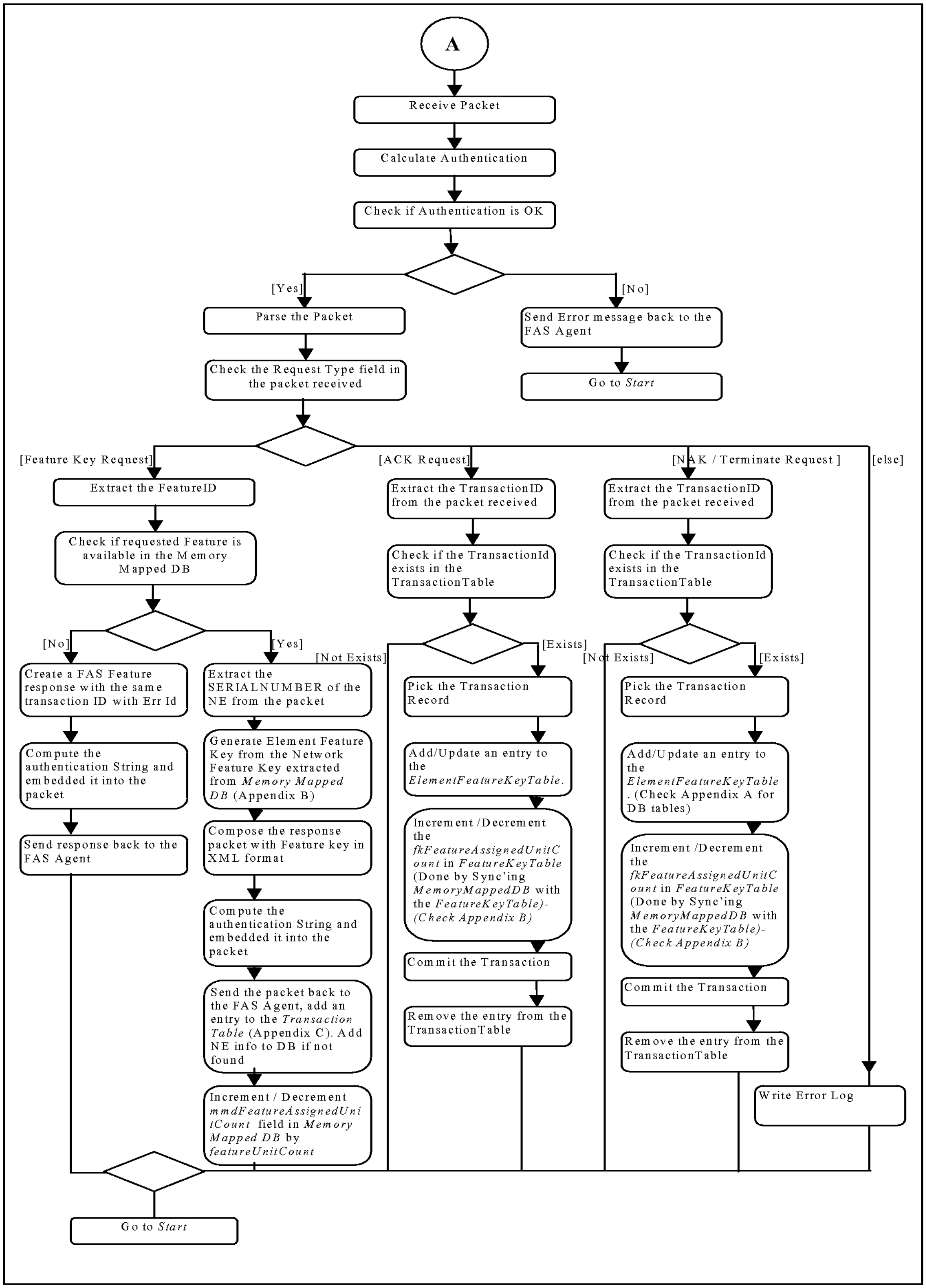

A
Receive Packet
Calculate Authentication
Check if Authentication is OK
[Yes]
[No]
Parse the Packet
Send Error message back to the FAS Agent
Check the Request Type field in the packet received
Go to Start
[Feature Key Request]
[ACK Request]
[NAK / Terminate Request]
[else]
Extract the FeatureID
Extract the TransactionID from the packet received
Extract the TransactionID from the packet received
Check if requested Feature is available in the Memory Mapped DB
Check if the TransactionId exists in the TransactionTable
Check if the TransactionId exists in the TransactionTable
[No]
[Yes]
[Not Exists]
[Exists]
[Not Exists]
[Exists]
Create a FAS Feature response with the same transaction ID with Err Id
Extract the SERIALNUMBER of the NE from the packet
Pick the Transaction Record
Pick the Transaction Record
Compute the authentication String and embedded it into the packet
Generate Element Feature Key from the Network Feature Key extracted from Memory Mapped DB (Appendix B)
Add/Update an entry to the ElementFeatureKeyTable.
Add/Update an entry to the ElementFeatureKeyTable . (Check Appendix A for DB tables)
Send response back to the FAS Agent
Compose the response packet with Feature key in XML format
Increment /Decrement the fkFeatureAssignedUnitCount in FeatureKeyTable (Done by Sync'ing MemoryMappedDB with the FeatureKeyTable)- (Check Appendix B)
Increment /Decrement the fkFeatureAssignedUnitCount in FeatureKeyTable (Done by Sync'ing MemoryMappedDB with the FeatureKeyTable)- (Check Appendix B)
Compute the authentication String and embedded it into the packet
Commit the Transaction
Commit the Transaction
Send the packet back to the FAS Agent, add an entry to the Transaction Table (Appendix C). Add NE info to DB if not found
Remove the entry from the TransactionTable
Remove the entry from the TransactionTable
Write Error Log
Increment / Decrement mmdFeatureAssignedUnitCount field in Memory Mapped DB by featureUnitCount
Go to Start

Figure 8 Activity Diagram for FASClientConnectionThread

6.7 CLIThread Component

6.7.1 Purpose

Command Line Interface Thread displays prompt to the operator. It is responsible for accepting and executing commands entered at the FAS prompt. CLI thread is launched after the startup of the FAS server.

6.7.2 Requirements/Functionality

The functionality and requirements implemented in this component are

- FAS Server shall support Command Line Interface (CLI) (Requirement 5.1.13)
- Providing a command line interface to FAS server
- Providing the status of feature activations

6.7.3 Input/Output

The following commands are supported through command line interface

1. *loadKeys*

   This command is used to force the loading of feature keys into FAS server once the feature key file is transferred to the respective directory. In normal scenarios FAS server picks up the feature key files to load the network feature keys at a regular interval.

2. *unloadKeys*

   This command unloads the features that are stored in the DB and Memory Mapped DB. This command removes Feature Keys only if it is not yet been assigned to any NE.

3. *listFeatures*

   This command is used to list the feature details that are available with the FAS server. Basically it gives the mapping between the feature IDs and their description. Some commands like list_availability and key_usage_details require feature ID as an argument.

4. *list_availability*

   This command displays the availability of the feature units for the feature ID specified in the arguments.

5. *key_usage_details*

   This command displays the details of the network elements (NMC chassis serial number, NE name, etc) that are presently has the feature activated by this FAS server for the feature ID specified in the argument.

6. *help/?*

   This command displays the commands available to use with this interface and brief description of each command.

7. *stop/exit*

   This command stops the FAS server.

6.7.4 Processing

CLIThread processing is shown in diagram below.

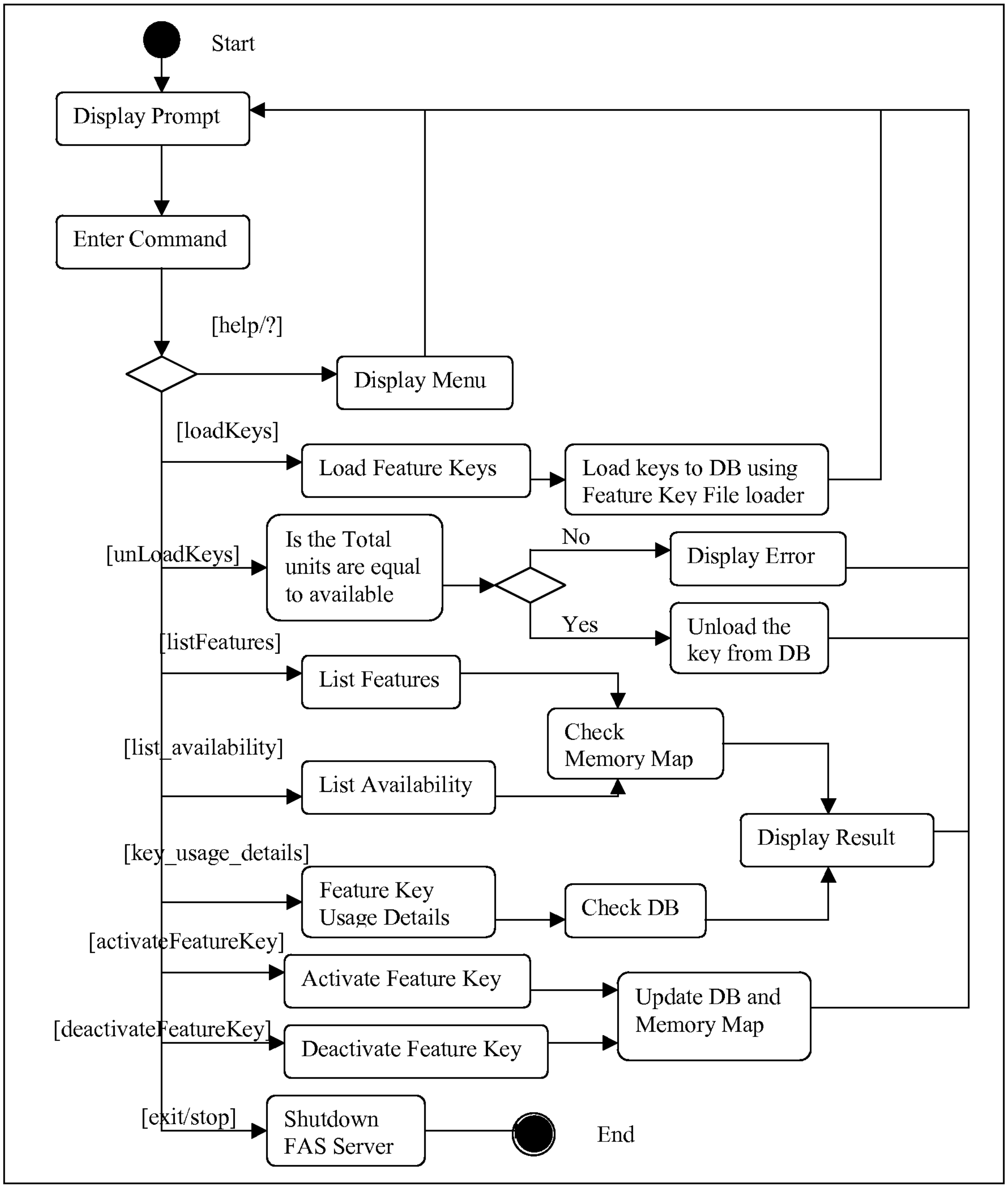


Figure 9 Activity Diagram for CLI Thread

6.8 FAS Feature File Loader Thread Component

6.8.1 Purpose

FAS feature file loader thread checks for the feature key files at the configured location at a regular time interval.

6.8.2 Requirements/Functionality

The functionality and requirements implemented in this component are

- Loading Network Feature Keys into the FAS Server via Feature Key Files. All the feature Keys Files shall be loaded from a specified directory (Requirement 5.1.4)
- The FAS Server shall support a log. (Requirement 5.1.11)
- The FAS Server shall keep track in persistent storage of every Feature Key received from CommWorks that is valid for that FAS Server. (Requirement 5.1.5)
- The FAS Server shall check the Feature Keys for duplication. (Requirement 5.1.6)
- The FAS Server should be able to revoke Feature Keys. (Requirement 5.1.7)
- The FAS Server should be able to re-use Feature Keys. (Requirement 5.1.8)
- The FAS Server shall verify the Network Feature Keys and authenticate the Feature Key Generator as the originator. (Requirement 5.1.12)
- Check for new feature key file in the feature keys directory
- If a key file exists, parse the file and verify the completion of the file. This is necessary as it is always possible to read the partial file, which is in the middle of transfer.
- If the file is complete, but the format is incorrect then the feature key file is moved to error directory and the same is logged.
- If file is parsed correctly the feature keys will be loaded into FAS server after proper authentication. The feature key files that are successfully parsed are moved to loaded directory for future reference.

6.8.3 Input/Output

Feature Key File (Input) (Check Appendix D)

6.8.4 Processing

Feature File Loader thread processing is shown in diagram below.

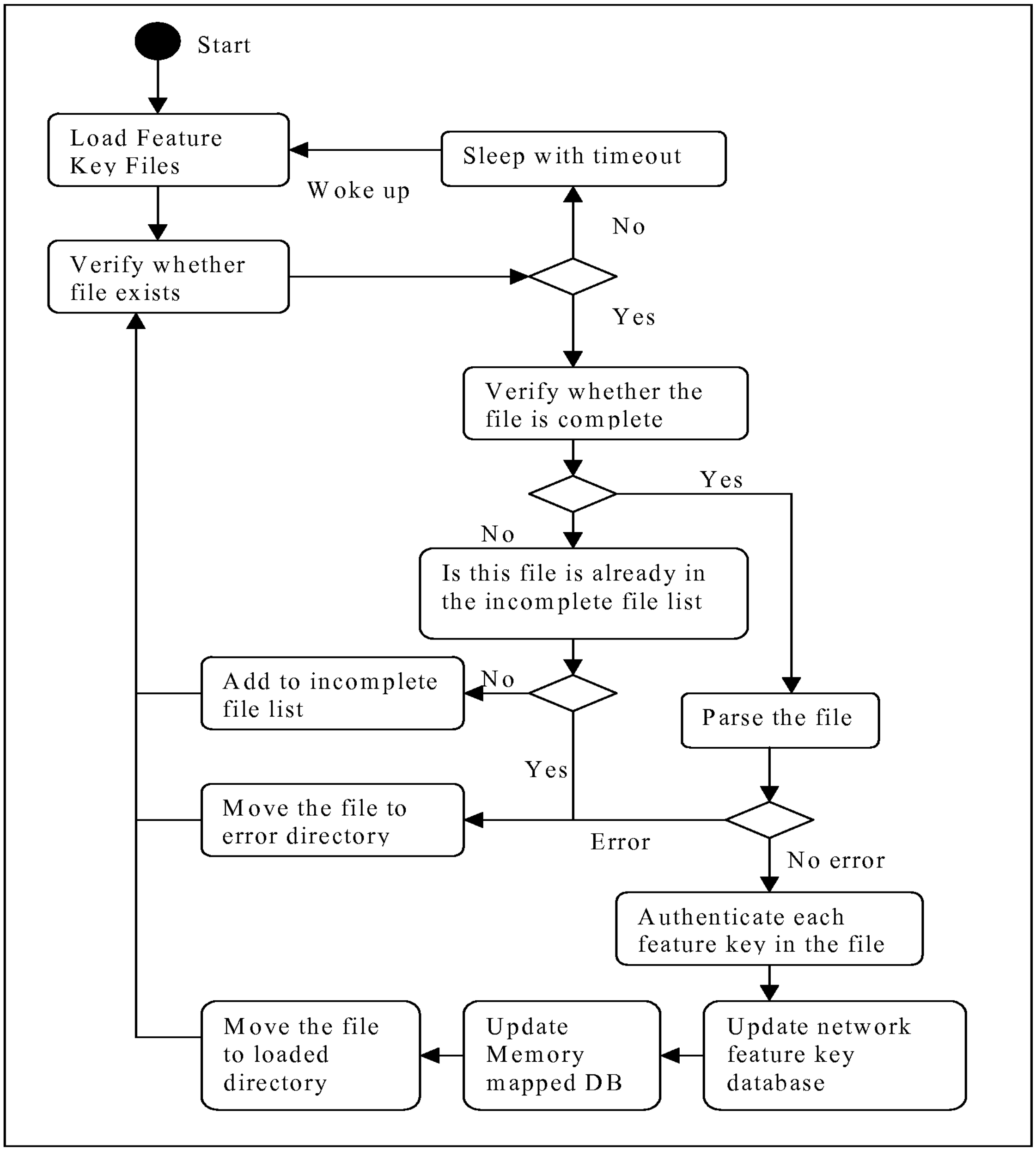

Start
Load Feature Key Files
Sleep with timeout
Woke up
No
Verify whether file exists
Yes
Verify whether the file is complete
Yes
No
Is this file is already in the incomplete file list
Add to incomplete file list
No
Parse the file
Yes
Move the file to error directory
Error
No error
Authenticate each feature key in the file
Move the file to loaded directory
Update Memory mapped DB
Update network feature key database

Figure 10 Activity Diagram of Feature Key File Loader Thread

7. APPENDIX A – DB TABLES

7.1 FeatureKeyTable

This table contains Network Feature Keys

Field Name	Field Type	Field Description
fkFeatureID	INTEGER	Specifies the Feature on a specific target component (Primary Key)
fkFeatureKeyVersion	TEXT	A string that indicates the structure version to facilitate -- backward compatibility if changes are made in the future
fkFeatureKeyType	TEXT	Indicates scope of FeatureKey i.e.: [GLOBAL | NETWORK | ELEMENT]
fkFeatureDescription	TEXT	A textual description of the feature.
fkFeatureTotalUnitCount	INTEGER	Total Number of Feature Units specified by this Feature Key
fkFeatureAssignedUnitCount	INTEGER	Number of units assigned to NEs till now. Feature request for this featureId cannot be processed if the featureTotalUnitCount = featureAssignedUnitCount
fkCustomerID	TEXT	CW ID of customer who was assigned the Feature Units in this Feature Key
fkfeatureUnitDuration	INTEGER	Number of seconds after activation the Feature Unit is valid and active

7.2 UpdateHistoryTable

This table contains the update history for each FeatureID

Field Name	Field Type	Field Description
uhFeatureID	INTEGER	CommWorks Unique ID. This is a foreign key from *FeatureKeyTable (fkFeatureID*)
uhFeatureUnitAdded	INTEGER	No. of feature Unit added in this update
uhFeatureKeySerialNumber	TEXT	CommWorks Unique ID, generated by Feature Key Generator.

7.3 ElementFeatureKeyTable

This table contains the entire Element feature Key details assigned to each NEs.

Field Name	Field Type	Field Description

ekFeatureID	INTEGER	CommWorks Unique ID. This is a foreign key from *FeatureKeyTable (fkFeatureID)*
ekElementFeatureKeySerialNumber	INTEGER	Generated by FAS Server
ekNeSerialNumber	TEXT	Unique Serial number received in Feature Key Request packet. This is a Foreign key from *NETable*
ekFeatureUnitCount	INTEGER	Total Number of Feature Units given to the NE

7.4 NETable

This table contains list of NEs that extracted Feature Keys from the FAS Server.

Field Name	**Field Type**	**Field Description**
neSerialNumber	TEXT	Unique Serial number received in Feature Key Request packet.
neIpAddress	TEXT	IpAddress of the NE
neName	TEXT	NE name.

8. APPENDIX B – MEMORY MAPPED DB

There are two memory-mapped tables one for the network feature keys and the other for element feature keys.

8.1 Network Memory Mapped DB

Field Name	**Field Type**	**Field Description**
mmdFeatureKeySerialNumber	INTEGER	CommWorks Unique ID, generated by Feature Key Generator
mmdFeatureID	INTEGER	Specifies the Feature on a specific target component
mmdFeatureKeyVersion	INTEGER	A string that indicates the structure version to facilitate -- backward compatibility if changes are made in the future
mmdFeatureKeyType	TEXT	Indicates scope of FeatureKey i.e.: [GLOBAL | NETWORK | ELEMENT]
mmdFeatureTotalUnitCount	INTEGER	Total Number of Feature Units specified by this Feature Key
mmdFeatureAssignedUnitCount	INTEGER	Number of units assigned to Nes till now. Feature request for this featureId cannot be processed if the mmdFeatureTotalUnitCount = mmdFeatureAssignedUnitCount
mmdFeatureDescription	TEXT	A textual description of the feature.
mmdfeatureUnitDuration	INTEGER	Number of seconds after activation the Feature Unit is valid and active
mmdCustomerID	TEXT	CW ID of customer who was assigned the Feature Units in this Feature Key

8.2 Element Memory Mapped DB

Field Name	**Field Type**	**Field Description**
emdNeSerialNumber	TEXT	Unique Serial number received in Feature Key Request packet. This is a Foreign key from *NETable*
emdFeatureID	INTEGER	CommWorks Unique ID. This is a foreign key from *FeatureKeyTable (fkFeatureID)*
emdFeatureUnitCount	INTEGER	Total Number of Feature Units given to the NE

9. APPENDIX C – TRANSACTION TABLE

Transaction Table used in FASClientConnectionThread.

Field Name	Field Type	Field Description
ttTransactionId	INTEGER	Unique transactionId generated by FAS Agent when sending a feature Key request to FAS Server
ttTimeout	INTEGER	Contains the timeout value
ttFeatureKeySerialNumber	INTEGER	This field is the index to the Memory Mapped DB
ttElementFeatureKeySerialNumber	INTEGER	The Serial number generated by FAS Server and sent to FAS Agent
ttNeSerialNumber	TEXT	Identification of NE
ttFeatureTotalUnitCount	INTEGER	Total Number of Feature Units specified by this Feature Key
ttDiffUnitCount	INTEGER	Difference of the units requested and units already used.

10. APPENDIX D - FEATURE KEY FILE

Feature Key File format from FAS Feature Key Generator is shown below

```
<? xml version="1.0" encoding=utf-8 ?>

<!-This file contains the structure for a CommWorks Feature Activation     -->
<!-Copyright XXXXXXX CommWorks Corp.  A wholly owned subsidiary of 3Com corp. -->
<!-All rights reserved.  May not be duplicated without permission          -->

<!-- Mandatory TAG -->
<FeatureKeyFile>


<!-- Optional ATTRIBUTE -->
<featureKeyFileVersion>
 1.0
</featureKeyFileVersion>


<!-- Mandatory TAG -->
<FeatureKeyList>
   <!-- One or more feature structures may exist within the FeatureKeyList -->


   <!-- Mandatory TAG -->
   <FeatureKey>


      <!-- Mandatory ATTRIBUTE -->
      <featureKeySerialNumber>
      </featureKeySerialNumber>


      <!-- Mandatory ATTRIBUTE -->
      <featureKeyVersion>
         1.0
      </featureKeyVersion>


      <!-- Mandatory ATTRIBUTE -->
      <featureKeyType>
      </featureKeyType>


      <!-- Mandatory ATTRIBUTE -->
      <featureID>
      </featureID>


      <!-- Mandatory ATTRIBUTE -->
      <featureDescription>
      </featureDescription>


      <!-- Mandatory ATTRIBUTE -->
      <featureUnitCount>
      </featureUnitCount>
```

```
        <!-- Mandatory ATTRIBUTE -->
        <featureUnitDuration>
        </featureUnitDuration>

        <!-- Optional ATTRIBUTE -->
        <customerID>
        </customerID>

        <!-- Mandatory ATTRIBUTE -->
        <destinationNodeIDType>
        </destinationNodeIDType>

        <!-- Mandatory ATTRIBUTE -->
        <destinationNodeID>
        </destinationNodeID>

        <!-- Optional  ATTRIBUTE -->
        <cwProxySignature>
          -- for future FAS System releases
        </cwProxySignature>

       <!-- Mandatory ATTRIBUTE -->

        <signatureSPI>
        </signatureSPI>

        <!-- Mandatory ATTRIBUTE -->
        <signature>
        <signature>

    <!-- Mandatory TAG -->
    </FeatureKey>


    <!-- One or more Feature Keys are supported within the FeatureKeyList -->

    <FeatureKey>
    </FeatureKey>

<!-- Mandatory TAG -->
</FeatureKeyList>


<!-- Mandatory TAG -->

</FeatureKeyFile>
```

11. APPENDIX E – FAS PROTOCOL

11.1 Header Information

Protocol Version (1 byte)	Packet Type (1 byte)	Packet Length (2 bytes)

11.1.1 Header Information Description

Field Name	Description
Protocol Version	This indicates the protocol version used. This shall be a simple positive integer. Valid values are from 1 to 255. Initial version is 1.
Packet Type	This is a 1-byte field. The valid values are from 1 to 255. Valid Values of Packet Type: • FAS Feature Request (1) • FAS Feature Response (2) • FAS Feature ACK (3) • FAS Feature NAK (4) • FAS Terminate (5) More packet types may be added if needed.
Packet Length	This is the total length of packet in number of bytes including the header information.

11.2 Tag Details

Tag (1 byte)	Length (2 bytes)	Value (variable)

11.2.1 Tags

Tag Name	Tag Numbers	Length (Byte/s)	Tag data Type (Interpreted as)
Packet Subtype	1	4	Integer
Transaction Id	2	4	Integer
Feature Identifier	3	4	Integer
Total Feature Units	4	4	Integer
Packet Response Code	5	4	Integer
Entity ID	6	?	Byte Array
Packet Authenticator	7	16	Byte Array
Auth SPI	8	4	Integer
Element Feature Key Number	9	4	Integer
Element Feature Key	10	?	Byte Array

11.3 Packet Type-Packet Subtype Matrix

Packet Type	Packet Subtype	Protocol Endpoints
FAS Feature Request (1)	1	FAS Agent and FAS Server
FAS Feature Request (1)	2	FAS Subagent and FAS Agent
FAS Feature Response (2)	1	FAS Agent and FAS Server
FAS Feature Response (2)	2	FAS Subagent and FAS Agent
FAS Feature ACK (3)	1	FAS Agent and FAS Server
FAS Feature ACK (3)	2	FAS Subagent and FAS Agent
FAS Feature NAK (4)	1	FAS Agent and FAS Server
FAS Feature NAK (4)	2	FAS Subagent and FAS Agent

11.4 FAS Feature Request Packet

The feature request packet contains the following fields in the form of tag-length-value.

Packet Subtype	The packet subtype indicates the parameter set contained in the message.
Packet Authenticator	Authenticator String
Auth SPI	Specifies which key and algorithm shall be used for authentication
Transaction Identifier	Unique value to identify packet for a complete transaction. This is generated by FAS-Agent for the Feature Request Packet only. All other kinds of packets in this transaction will use the same packet identifier as in feature request packet.
Entity ID	Serial Number of the Total Control 2000 Control Shelf. This is unique for each Total Control 2000 system.
Feature Identifier	The identifier of the feature being requested.
Total Feature Units	Total number of feature units of a particular feature identifier being requested.

11.5 FAS Feature Response Packet

The feature response packet contains the following fields in the form of tag-length-value.

Packet Subtype	The packet subtype indicates the parameter set contained in the message.
Packet Authenticator	Authenticator String
Auth SPI	Specifies which key and algorithm shall be used for authentication
Transaction Identifier	Unique value to identify packet for a complete transaction. This is the same value as in the feature request packet.
Feature Identifier	The identifier of the feature being requested.

Total Feature Units	Total number of feature units of a particular feature identifier being requested.
Packet Response Code	This contains the error code for the protocol response packet. Valid values include: • No Error • Authentication Failed • Protocol Version Mismatch • Feature Units not available Other values may be added when need be.
Feature Key in XML format	Feature Key in XML format It shall be parsed by the FAS-Agent.

11.6 FAS Feature Ack Packet

The feature ack packet contains the following fields in the form of tag-length-value.

Packet Subtype	The packet subtype indicates the parameter set contained in the message.
Packet Authenticator	Authenticator String
Auth SPI	Specifies which key and algorithm shall be used for authentication
Transaction Identifier	Unique value to identify packet for a complete transaction. This is the same value as in the feature response packet.
Packet Response Code	This contains the error code for the protocol response packet. Valid values include: • No Error

11.7 FAS Feature Nak Packet

The feature nak packet contains the following fields in the form of tag-length-value.

Packet Subtype	The packet subtype indicates the parameter set contained in the message.
Packet Authenticator	Authenticator String
Auth SPI	Specifies which key and algorithm shall be used for authentication
Transaction Identifier	Unique value to identify packet for a complete transaction. This is the same value as in the feature response packet.
Packet Response Code	This contains the error code for the protocol response packet. Valid values include: • Authentication Failed • Protocol Version Mismatch Other values may be added when need be.

11.8 FAS Feature Terminate Packet

The feature terminate packet contains the following fields in the form of tag-length-value.

Packet Authenticator	Authenticator String
Auth SPI	Specifies which key and algorithm shall be used for authentication
Transaction Identifier	Unique value to identify packet for a complete transaction. This is the same value as in the feature response packet.

12. APPENDIX F – FAS SERVER MIB

```
--***************************************************************************
--*     $Revision:   $
--*     $Date:       $
--*
--*  3COM FASS-MIB : FAS Server MIB
--*
--***************************************************************************

FASS-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY, IpAddress, Counter32,
Integer32
            FROM SNMPv2-SMI
        DateAndTime, TEXTUAL-CONVENTION
            FROM SNMPv2-TC
        MODULE-COMPLIANCE, OBJECT-GROUP
            FROM SNMPv2-CONF
        fasServer
            FROM CSG-DEFINITIONS-MIB;

      fasServerMIB MODULE-IDENTITY
        LAST-UPDATED " XXXXXXX"
        ORGANIZATION "Carrier Business Unit"
        CONTACT-INFO "Postal: CommWorks Corporate Offices
                              3800 Golf Road
                              Rolling Meadows, Illinois 60008
                              USA
                      Tel:    847-262-5000
                      Web:    http://www.commworks.com/"
        DESCRIPTION
            "The MIB module for the Feature Activation System Server
component."

        REVISION      " XXXXXXX"
        DESCRIPTION
            "The initial revision of this MIB module."

        REVISION      " XXXXXXX"
        DESCRIPTION
            "-Added Textual convention for FasFeatureId
             -Changed naming convention for FAS Server Performance
             attributes"

        ::= { fasServer 1 }

      -- Textual Conventions

      -- For FAS Feature ID
      FasFeatureID ::= TEXTUAL-CONVENTION
      STATUS      current
      DESCRIPTION
          "Uniquely identifies a feature that can be enabled."
      SYNTAX      Integer32 (0..2147483647)

      fasSMIBObjects OBJECT-IDENTITY
            STATUS current
            DESCRIPTION
            "FAS Server MIB Objects"
      ::= { fasServerMIB 1 }
```

```
fasSAttr   OBJECT-IDENTITY
      STATUS current
      DESCRIPTION
      "Attributes pertaining to FAS Server operation and performance"
::= { fasSMIBObjects 1 }

-- FAS Server Command Details

fasSCmd OBJECT-IDENTITY
      STATUS current
      DESCRIPTION
      "FAS Server Command Objects, to instruct the server to
       execute a function and get the result stored"
::= { fasSAttr 1 }

fasSCmdFunction OBJECT-TYPE
      SYNTAX INTEGER{
            nocommand(1),
            loadfeaturekey(2),
            unloadfeaturekey(3)
      }
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
      "This object contains the value which decsribe the command
      which is being invoked.
      Value List:
            nocommand(1) : None state

            loadfeaturekey(2) : instructs the server to load the
               feature key files from the FAS Server 'load' directory

            unloadfeaturekey(3) : instructs the server to unload
               the feature key. This function is executed only
               if the fasSFeatureUHFeatureUnits of the requested
               feature key (in Update History Table)
               is less than or equal to fasSNFeatureKeyTotalUnits
               (in Network Feature Key Table)"
::= { fasSCmd 1 }

fasSCmdParam OBJECT-TYPE
      SYNTAX OCTET STRING (SIZE(1..255))
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
      "This object contains parameters that are specific to the
       particular command being issued.  In some cases, there will
       be no additional parameters required"
::= { fasSCmd 2 }

fasSCmdStatus OBJECT-TYPE
      SYNTAX INTEGER{
            none(1),
            inprogress(2),
            success(3),
            failed(4),
            notsupported(5),
            aborted(6)
      }
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
      "This object contains the result of the most recently
      requested command or test, or the value none(1) if no
      commands have been requested since the last reset."
::= { fasSCmd 3 }

fasSCmdErrorCode OBJECT-TYPE
      SYNTAX INTEGER{
            none(1),
            invalidfeaturekey(2),
            nofilefound(3),
            erroredfile(4),
            invalidfunction(5),
            internalerror(6)
      }
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
      "This object gives error reason for the failure.
       Value List:

          none(1) : no error

          invalidfeaturekey(2) : This error occurs only for
             'unloadfeaturekey' function. This signifies
             that the featurekey given in fasSCmdParam is not
             a valid key.

          nofilefound(3) : This occurs only for 'loadfeaturekey'
             function. The error signifies that the 'load'
             directory is empty.

          erroredfile(4) : This signifies that some error
             files found while executing the 'loadfeaturekey'
             command.

          invalidfunction(5) : invalid function.

          internalerror(6) : Internal error. This error
             could happen for any function. Check server
             debug/server log for more details."
::= { fasSCmd 4 }

-- FAS Server Trap Details

fasSTrap OBJECT-IDENTITY
      STATUS current
      DESCRIPTION
      "FAS Server Trap object"
::= { fasSAttr 2 }


fasSTrapSeqNo OBJECT-TYPE
      SYNTAX Integer32 (0..2147483647)
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
      "This object increments every time a trap is generated by
      the FASServer.  This object is sent in the var-bind list of each
      trap and can be used by management stations to detect when
      a trap has been lost."
::= { fasSTrap 1 }

-- FAS Trap Destination Table
```

```
fasSTrapDestTable OBJECT-TYPE
      SYNTAX SEQUENCE OF FasSTrapDestEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
      "This table contains the trap destination details,
       where the traps has to be forwarded"
::= { fasSTrap 2 }

fasSTrapDestEntry OBJECT-TYPE
      SYNTAX FasSTrapDestEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
      "One row for each trap forward destination"
      INDEX { fasSTrapDestIp }
::= { fasSTrapDestTable 1 }

FasSTrapDestEntry ::=
SEQUENCE {
      fasSTrapDestIp                   IpAddress,
      fasSTrapDestPort                 Integer32,
      fasSTrapDestCommunityStr         OCTET STRING,
      fasSTrapDestSNMPVersion          Integer32,
      fasSTrapDestDescription          OCTET STRING
}

fasSTrapDestIp OBJECT-TYPE
      SYNTAX IpAddress
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
      "This object contains the trap destination
       IP address where the trap has to be forwarded"
::= { fasSTrapDestEntry 1 }

fasSTrapDestPort OBJECT-TYPE
      SYNTAX Integer32 (0..65535)
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
      "This object contains the trap destination port number"
::= { fasSTrapDestEntry 2 }

fasSTrapDestCommunityStr OBJECT-TYPE
      SYNTAX OCTET STRING (SIZE(1..32))
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
      "This object contains the SNMP Community string
       for the trap destionation."
::= { fasSTrapDestEntry 3 }

fasSTrapDestSNMPVersion OBJECT-TYPE
      SYNTAX Integer32 (1..3)
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
      "This object contains the SNMP Version of the trap destionation"
::= { fasSTrapDestEntry 4 }

fasSTrapDestDescription OBJECT-TYPE
```

```
            SYNTAX OCTET STRING (SIZE(1..32))
            MAX-ACCESS read-write
            STATUS current
            DESCRIPTION
            "This object contains a plain text description of the
             trap destination to which this entry pertains."
      ::= { fasSTrapDestEntry 5 }


      -- FAS Server Performance Attributes

      fasSPAttr OBJECT-IDENTITY
            STATUS current
            DESCRIPTION
            "FAS Server Performance Objects"
      ::= { fasSAttr 3 }

      fasSPAttrStartTime OBJECT-TYPE
            SYNTAX DateAndTime
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "This object contains the FAS Server Start time."
      ::= { fasSPAttr 1 }

      fasSPAttrAuthFailures OBJECT-TYPE
            SYNTAX Counter32
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "This object is the count of Authentication failures
             in FAS Server."
      ::= { fasSPAttr 2 }

      fasSPAttrMalformedPackets OBJECT-TYPE
            SYNTAX Counter32
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "This object is the count of Severely errored packets,
             Which the FAS Server will siliently discard the packet"
      ::= { fasSPAttr 3 }

      fasSPAttrConnectFailures OBJECT-TYPE
            SYNTAX Counter32
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "This object is the count of Connect request that are
             dropped, because of unavailability of Request Handler
             thread resource."
      ::= { fasSPAttr 4 }

      fasSPAttrNEsServed OBJECT-TYPE
            SYNTAX Integer32 (1..2147483647)
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "This object is the count of Network Element(NE) served by FAS
Server
             till date."
      ::= { fasSPAttr 5 }
```

```
-- Feature Key Related Tables

fasSFeatureKeyDetails OBJECT-IDENTITY
      STATUS current
      DESCRIPTION
      "FAS Server Feature Key related attributes"
::= { fasSMIBObjects 2 }

-- Network Feature Key Table

fasSNFeatureKeyTable OBJECT-TYPE
      SYNTAX SEQUENCE OF FasSNFeatureKeyEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
      "This table contains all the network Feature Key
       currently supported by the FASServer"
::= { fasSFeatureKeyDetails 1 }

fasSNFeatureKeyEntry OBJECT-TYPE
      SYNTAX FasSNFeatureKeyEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
      "This row contains information pertaining to a Network
       Feature Id."
      INDEX { fasSNFeatureKeyId }
::= { fasSNFeatureKeyTable 1 }

FasSNFeatureKeyEntry ::=
SEQUENCE {
      fasSNFeatureKeyId               FasFeatureID,
      fasSNFeatureKeyVersion          Integer32,
      fasSNFeatureKeyType             Integer32,
      fasSNFeatureKeyDescription      OCTET STRING,
      fasSNFeatureKeyTotalUnits     Integer32,
      fasSNFeatureKeyAssignedUnits  Integer32,
      fasSNFeatureKeyCustomerID       OCTET STRING,
      fasSNFeatureKeyDurationUnit     Integer32
}

fasSNFeatureKeyId OBJECT-TYPE
      SYNTAX FasFeatureID
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
      "An Id to uniquely identify a Feature across Commworks
       products line."
::= { fasSNFeatureKeyEntry 1 }

fasSNFeatureKeyVersion OBJECT-TYPE
      SYNTAX Integer32 (1..2147483647)
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
      "Version number of this feature Key, since each key has
       version number associated to it."
::= { fasSNFeatureKeyEntry 2 }

fasSNFeatureKeyType OBJECT-TYPE
      SYNTAX INTEGER{
            global(1),
            network(2),
                  element(3)
            }
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "Indicates scope of FeatureKey i.e.: [GLOBAL | NETWORK |
ELEMENT]."
      ::= { fasSNFeatureKeyEntry 3 }

      fasSNFeatureKeyDescription OBJECT-TYPE
            SYNTAX OCTET STRING (SIZE(1..255))
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "This object describes the feature"
      ::= { fasSNFeatureKeyEntry 4 }

      fasSNFeatureKeyTotalUnits OBJECT-TYPE
            SYNTAX Integer32 (0..2147483647)
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "Total number of feature units available for this feature ID."
      ::= { fasSNFeatureKeyEntry 5 }

      fasSNFeatureKeyAssignedUnits OBJECT-TYPE
            SYNTAX Integer32 (0..2147483647)
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "Number of feature units assigned to Network Element(NE).
Remaining
             feature units for usage is the difference between
             TotalUnits & AssignedUnits"
      ::= { fasSNFeatureKeyEntry 6 }

      fasSNFeatureKeyCustomerID OBJECT-TYPE
            SYNTAX OCTET STRING (SIZE(1..32))
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "A Unique Customer Id generated by Commworks"
      ::= { fasSNFeatureKeyEntry 7 }

      fasSNFeatureKeyDurationUnit OBJECT-TYPE
            SYNTAX Integer32 (0..2147483647)
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "The Time this Feature expires. This will always
             be 0 for this release"
      ::= { fasSNFeatureKeyEntry 8}

      -- Feature Key Update History Table

      fasSFeatureUHTable OBJECT-TYPE
            SYNTAX SEQUENCE OF FasSFeatureUHEntry
            MAX-ACCESS not-accessible
            STATUS current
            DESCRIPTION
            "This table contains update history for every featureId
             found in fasSNFeatureKeyTable"
      ::= { fasSFeatureKeyDetails 2 }
      fasSFeatureUHEntry OBJECT-TYPE
            SYNTAX FasSFeatureUHEntry
            MAX-ACCESS not-accessible
            STATUS current
            DESCRIPTION
            "A row contains the feature key details on how many
             feature unit count this feature key contributed to
             the feature ID"
            INDEX { fasSFeatureUHFeatureKeySerialNo }
      ::= { fasSFeatureUHTable 1 }

      FasSFeatureUHEntry ::=
      SEQUENCE {
            fasSFeatureUHFeatureKeySerialNo   Integer32,
            fasSFeatureUHFeatureId            FasFeatureID,
            fasSFeatureUHFeatureUnits       Integer32
      }

      fasSFeatureUHFeatureKeySerialNo OBJECT-TYPE
            SYNTAX Integer32 (1..2147483647)
            MAX-ACCESS not-accessible
            STATUS current
            DESCRIPTION
            "A unique index generated by the FASKeyGenerator application while
             creating Network Feature Key"
      ::= { fasSFeatureUHEntry 1 }

      fasSFeatureUHFeatureId OBJECT-TYPE
            SYNTAX FasFeatureID
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "An Id to uniquely identify a Feature Key across Commworks
products. "
      ::= { fasSFeatureUHEntry 2 }

      fasSFeatureUHFeatureUnits OBJECT-TYPE
            SYNTAX Integer32 (1..2147483647)
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "Contains the number of feature Units added to the FeatureId by
             this updation."
      ::= { fasSFeatureUHEntry 3 }

      -- Element Feature Key Table

      fasSEFeatureKeyTable OBJECT-TYPE
            SYNTAX SEQUENCE OF FasSEFeatureKeyEntry
            MAX-ACCESS not-accessible
            STATUS current
            DESCRIPTION
            "This table contains all the Feature Keys assigned to every
Network Element(NE) in
             a FAS Domain"
      ::= { fasSFeatureKeyDetails 3 }

      fasSEFeatureKeyEntry OBJECT-TYPE
            SYNTAX FasSEFeatureKeyEntry
            MAX-ACCESS not-accessible
            STATUS current
            DESCRIPTION
            "Contains the feature key usage details for each Network
Element(NE). One row
             for each feature usage."
            INDEX { fasSEFeatureKeyNESerialNo,fasSEFeatureKeyId }
      ::= { fasSEFeatureKeyTable 1 }

      FasSEFeatureKeyEntry ::=
      SEQUENCE {
            fasSEFeatureKeyNESerialNo   OCTET STRING,
            fasSEFeatureKeyId           FasFeatureID,
            fasSEFeatureKeySerialNo     Integer32,
            fasSEFeatureKeyUnits      Integer32
      }

      fasSEFeatureKeyNESerialNo OBJECT-TYPE
            SYNTAX OCTET STRING (SIZE(1..32))
            MAX-ACCESS not-accessible
            STATUS current
            DESCRIPTION
            "Serial number of the Network Element(NE) - INDEX"
      ::= { fasSEFeatureKeyEntry 1 }

      fasSEFeatureKeyId OBJECT-TYPE
            SYNTAX FasFeatureID
            MAX-ACCESS not-accessible
            STATUS current
            DESCRIPTION
            "An Id to uniquely identify a Feature Key across Commworks
             products. - INDEX "
      ::= { fasSEFeatureKeyEntry 2 }

      fasSEFeatureKeySerialNo OBJECT-TYPE
            SYNTAX Integer32 (1..2147483647)
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "A unique index generated by the FAS Server while
             generating the Element feature key. This will always be '0'
             for this release"
      ::= { fasSEFeatureKeyEntry 3 }

      fasSEFeatureKeyUnits OBJECT-TYPE
            SYNTAX Integer32 (1..2147483647)
            MAX-ACCESS read-only
            STATUS current
            DESCRIPTION
            "Contains the number of feature Units assigned to
             the Network Element(NE) thro this Element Feature Key."
      ::= { fasSEFeatureKeyEntry 4 }

      -- Network Element Table

      fasSNetworkElementTable OBJECT-TYPE
            SYNTAX SEQUENCE OF FasSNetworkElementEntry
            MAX-ACCESS not-accessible
            STATUS current
            DESCRIPTION
            "This table contains all the Network Element(NE) supported by this
FASServer"
      ::= { fasSFeatureKeyDetails 4 }

      fasSNetworkElementEntry OBJECT-TYPE
            SYNTAX FasSNetworkElementEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
      "One row for each Network Element(NE) supported by this FASServer"
      INDEX { fasSNetworkElementSerialNo }
::= { fasSNetworkElementTable 1 }

FasSNetworkElementEntry ::=
SEQUENCE {
      fasSNetworkElementSerialNo   OCTET STRING,
      fasSNetworkElementIpAddress  IpAddress,
      fasSNetworkElementName       OCTET STRING
}

fasSNetworkElementSerialNo OBJECT-TYPE
      SYNTAX OCTET STRING (SIZE(1..32))
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
      "Serial number of the Network Element(NE). - INDEX"
::= { fasSNetworkElementEntry 1 }

fasSNetworkElementIpAddress OBJECT-TYPE
      SYNTAX IpAddress
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
      "Network Element(NE) Ip Address"
::= { fasSNetworkElementEntry 2 }

fasSNetworkElementName OBJECT-TYPE
      SYNTAX OCTET STRING (SIZE(1..32))
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
      "Network Element(NE) name if any."
::= { fasSNetworkElementEntry 3 }


-- Conformance Information
fasSMIBConformance OBJECT-IDENTITY
      STATUS current
      DESCRIPTION
      "FAS Server MIB Conformance"
::= { fasServerMIB 2 }

fasSMIBCompliances      OBJECT IDENTIFIER ::= { fasSMIBConformance 1 }
fasSMIBGroups           OBJECT IDENTIFIER ::= { fasSMIBConformance 2 }

-- Compliance Information

fasSMIBCompliance MODULE-COMPLIANCE
      STATUS   current
      DESCRIPTION
         "The compliance statement for FAS Server Implementation"
      MODULE -- This MIB module
      MANDATORY-GROUPS {
           fasSCmdGroup,
           fasSTrapControlGroup,
           fasSPerfGroup,
           fasSFeatureKeyInfoGroup
      }
::= { fasSMIBCompliances 1 }
```

```
      -- Units of Conformance

      fasSCmdGroup OBJECT-GROUP
            OBJECTS   {
                 fasSCmdFunction,
                 fasSCmdParam,
                 fasSCmdStatus,
                 fasSCmdErrorCode
            }
            STATUS   current
            DESCRIPTION
            "A collection of objects used to instruct FAS Server to execute a
function"
      ::= { fasSMIBGroups 1 }

      fasSTrapControlGroup OBJECT-GROUP
            OBJECTS {
                 fasSTrapSeqNo,
                 fasSTrapDestPort,
                 fasSTrapDestCommunityStr,
                 fasSTrapDestSNMPVersion,
                 fasSTrapDestDescription
            }
            STATUS  current
            DESCRIPTION
            "A collection of objects used for Notification purpose."
      ::= { fasSMIBGroups 2 }

      fasSPerfGroup OBJECT-GROUP
            OBJECTS {
                 fasSPAttrStartTime,
                 fasSPAttrAuthFailures,
                 fasSPAttrMalformedPackets,
                 fasSPAttrConnectFailures,
                 fasSPAttrNEsServed
            }
            STATUS current
            DESCRIPTION
            "A collection of FAS Server Performance attributes."
      ::= { fasSMIBGroups 3 }

      fasSFeatureKeyInfoGroup OBJECT-GROUP
            OBJECTS   {
                 fasSNFeatureKeyVersion,
                 fasSNFeatureKeyType,
                 fasSNFeatureKeyDescription,
                 fasSNFeatureKeyTotalUnits,
                 fasSNFeatureKeyAssignedUnits,
                 fasSNFeatureKeyCustomerID,
                 fasSNFeatureKeyDurationUnit,
                 fasSFeatureUHFeatureId,
                 fasSFeatureUHFeatureUnits,
                 fasSEFeatureKeySerialNo,
                 fasSEFeatureKeyUnits,
                 fasSNetworkElementIpAddress,
                 fasSNetworkElementName
            }
            STATUS   current
            DESCRIPTION
              "A collection of objects that gives information on Feature Key
               usage and availability"
      ::= { fasSMIBGroups 4}
```

```
END
```